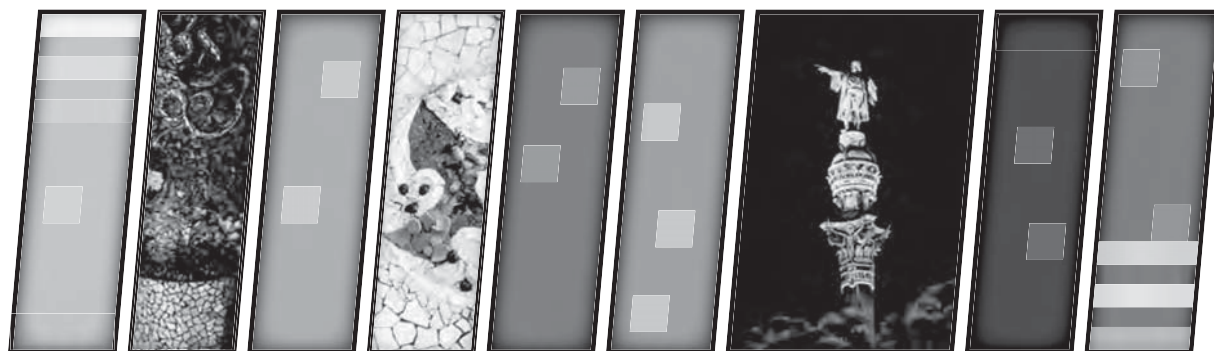


EDULEARN₁₆

**8TH INTERNATIONAL CONFERENCE
ON EDUCATION AND NEW LEARNING
TECHNOLOGIES**

**BARCELONA (SPAIN)
4TH - 6TH OF JULY, 2016**



CONFERENCE PROCEEDINGS

Published by
IATED Academy
iated.org

EDULEARN16 Proceedings
8th International Conference on Education and New Learning Technologies
July 4th-6th, 2016 — Barcelona, Spain

Edited by
L. Gómez Chova, A. López Martínez, I. Candel Torres
IATED Academy

ISBN: 978-84-608-8860-4
ISSN: 2340-1117
Depósito Legal: V-1421-2016

Book cover designed by
J.L. Bernat

All rights reserved. Copyright © 2016, IATED

The papers published in these proceedings reflect the views only of the authors. The publisher cannot be held responsible for the validity or use of the information therein contained.

TEACHING IMAGE AND VIDEO PROCESSING WITH A PRACTICAL CASES-BASED METHODOLOGY AT THE UNIVERSITY OF ALICANTE

P. Gil, G.J. García, S.T. Puente, C.M. Mateo, B. Alacid, D. Mira

University of Alicante (SPAIN)

Abstract

At universities, the courses about computer vision and image processing have been usually taught using a classical methodology based on textbooks, notes and practical exercises which are explained on a blackboard, although sometimes the teachers use video projector and PC presentations for showing slides with static contents; mainly slides with text, figures and photos. However, this teaching model is focused in teacher instead of students and consequently, this approach is not effective when the teacher seeks to achieve cognitive objectives which involve a student's critical thinking. That is, student should be able to develop skills such as acquired knowledge about how filters, operators, methods and techniques for image processing are implemented by computer vision software libraries. Besides, student should be motivated to use that background in order to solve practical and real problems in which it is advisable both a study and an analysis of features based on image processing.

This manuscript presents the development, implementation and assessment of a practical-cases-based engineering course, concretely in the Image and Video Processing degree course at the University of Alicante. All course lectures and hands-on laboratory activities have as the main goal that student acquires not only image-specific technical skills but also general knowledge in data analysis to discover phenomena in pixel regions of images and video frames. This way, our teaching-learning process accomplishes both knowledge assimilation and skill developments using a strategy continuous evaluation. In addition, this work presents the progress of student results as well as a comparison between the assessments and scores of students in this year and past years in issues related to specific subject.

The Image and Video Processing course is a compulsory subject of Degree in Multimedia Engineering at the University of Alicante in Spain. It is taught during 14-week period of the third academic year, also it has a workload of 6 ECTS (European Credit Transfer and Accumulation System) distributed as 30 hours of computer and laboratory practices and 30 hours of theory class. Nowadays, 80 students are enrolled in this course.

Keywords: Engineering Education, Image processing, Practical cases and projects-based learning, Continuous assessment.

1 INTRODUCTION

1.1 Education in Image Processing and Computer Vision

There are differences between concepts as Image Processing (IP) and Computer Vision (CV). On the one hand, IP includes the methods and algorithms about filtering, noise removal, image enhancement, feature extraction such as corner and edge detection, colour processing based on thresholding or segmentation and so forth. In IP techniques, both input and output are images. On the other hand, CV refers to methods and algorithms which emulate human vision and they are used to obtain quantitative and qualitative information from an input image such as detection and recognition of objects, people or scenes, characteristic extraction of them, descriptors for identifying, measurements and so forth. Thus, CV techniques return as output numerical values which represent measurements or descriptions of the items registered in the image. That is, the output is the knowledge of the scene registered within the image. CV uses always IP techniques to transform the input image. Besides, Video Processing uses both types of techniques seeking to optimize the processing so that it is not done frame by frame or using the entire image. This way, it is possible to achieve quasi real-time processing.

During the last two decades, IP and CV have progressively been incorporated as subjects into both undergraduate and graduate programs of some engineering studies. The first relevant study focused on the education of these new subjects was presented in [1] where various approaches to teaching

them were reviewed. The methodology used for teaching these subjects is depending on the specific knowledge of the students in these topics as well as specific knowledge in others contents, such as mathematics and programming languages. Specially, this last contents determines the kind of technological tools which should be proposed for the understanding of topics of IP and CV. Thus, sometimes authors have proposed interfaces to test the methods and algorithms without implementing code. But also authors made proposal which needed the implementation of small scripts with small code lines using well-known programming languages as Matlab or C/C++. This depends on student skills, if they are secondary-level, undergraduate or graduate, also in which engineering or science course they are enrolled.

In university environments, the new tendency is to learn IP and CV by means of practical-cases looking for the solution of real problems and the increase of the student motivation. Therefore, the laboratory practical cases must have a significant and relevant workload as in [2]. Some authors have reported using more student-centered approaches in teaching IP and CV, such as problem-based and project based learning [3] and [4], instead of using lecture-based teaching methods. In [5], students learn basic concepts such as colour spaces or image transforms while they are developing a vision-based game controller as interface human-machine for game consoles. In [6] is described a course which makes use of a project-oriented design of traffic sign recognition and a project-based laboratory teaching.

1.2 Tools for learning Image Processing and Computer Vision

In the literature, there are several environments and tools to learn concepts of Image and Video Processing (IVP) and test methods and techniques as well as design and implement own algorithms. All of them can be applied to learn computer vision or image processing in university degree courses [7] or non-academic environments [8] such as industry representative. A review comparing several tools was presented in [1]. However, the selection depends on the prior knowledge and skills acquired by students in previous courses. The tools for learning image processing and computer vision can be classified with respect to the programming knowledge level. Therefore, the tools based on programming can be classified using as criteria the programming language. The most used programming languages for image processing are C/C++, Matlab, Java and Python. Imlab [9], RAVL [10], libCVD [11] and OPENCV [8] are free-open libraries often used to teach and learn computer vision more than image processing in computer engineering degree or others in which students have a deep knowledge in programming techniques. The first three are built in C/C++ and the last, OPENCV, is built in C but it supports Java and Python. The tools based on Java or Python usually require less programming skill due to the level of encapsulation is greater than the C/C++ and also memory problems are easier to manage.

Furthermore, there are some works aimed at creating an intuitive and friendly framework for students using C or Java and thus, the requirements of programming skills can be attenuated. Non-commercial solutions based on this idea, it should be noted IP-Lab and ImageJ [12], TITERE [13] and more recently, NeatVision [14] and JavaVis [15], each one of them built in Java. Along this line, there are other solutions based on C such as VISUAL [16] and AVS [17]. Note that the pioneer was TITERE which describes a workspace to handle a real image acquisition system (it can control camera and an illumination system) and process its images from a remote environment using internet. All of them are frameworks in which students chose different blocks that represent basic algorithms and connect them to perform an image processing algorithm more complex. The same idea has been exported later in NetVision, JavaVis and AVS, the programming by connecting modules. The important is that all of them allow students learn basic concepts and algorithms of image processing and computer vision without require that they have any experience in low-level programming. Although, these tools do not always allow students to create new modules writing code C or Java, supposing that this was necessary to do it. Besides, both TITERE and AVS implement an image acquisition module, the rest solely works with images stored on disk or memory.

The three main drawbacks of all these tools are: the limitation on the number and kind of both operations and implemented algorithms, the difficulty of adding new methods and algorithms using code to low-level from the code of other algorithms pre-programmed within the tool as well as the capability for sequence video processing and not solely static images. In these cases, there are three practical solutions: the low-level programming with free-open libraries as OpenCV; the use of tools whose architecture is based on this kind of libraries such as IPLT [18], Raydiance [19] or iceWing [20]; or the use of commercial software which integrate these features but they are easier to program and

they do not require use very low-level code as MATLAB [21] with specific toolbox [22] such as Image Processing Toolbox and Computer Vision Toolbox or LabView [23] among many others.

In the first group, IPLT is a high level interface designed as open-source and whose architecture was implemented as a hybrid of C++ and Python. Raydiance is a user interface for controlling the parameters of computer vision algorithms implemented with OpenCV and libCVD. IceWing is a graphical plugin shell optimized in order to control of the parameterization of algorithms and improve the communication among them. It also includes communication and handling from other external libraries, for example OpenCV or RAVL. Another example of vision-related graphical interface is Visual-VIPERS [24]. Like the others, this tool allows students to gain more hands-on experience developing imaging algorithm for vision applications while student is being isolated of the implementation but giving the possibility of studying how the parameters of methods affect to the solution. In the second group, MATLAB is a software that has a high-performance for both matrix and vector computation and visualization of graphs. Consequently, MATLAB has been widely used for image processing in undergraduate, graduate and postgraduate programs as is shown in [7], [25] and [26].

Summarizing, the courses on computer vision are usually offered to the students from non-academic environments such as industry representatives in order to provide engineering students both theoretical and hands-on practical experience. Consequently, the educational and research communities have presented many software packages and tools aimed at learning topics of image processing and computer vision in various fields ranging from science to engineering. The idea always is to learn to extract structural information from images. Thereby, frequently educators and researchers create specific tools. For example, IPLT was specifically created for image processing of electron microscope images in science fields or another tool called CoLFDImap [27] was developed for teaching of FPGA-based digital image processing in undergraduate courses in computer engineering programs. But at other times, the educators have decided to use commercial software because students require use real equipment of machine vision and optical devices, also the short duration and specific topics addressed in the course forces that the teaching-learning is based on examples and practical cases to solve a real simple problem of the industry. Thus, MATLAB and Labview have been used in [28] in courses of a non-academic environment.

2 METHODOLOGY AND FRAMEWORK

The course presents in this paper is the IVP course which is a compulsory subject of Degree in Multimedia Engineering at the University of Alicante (UA) in Spain. Its main objective is to train future information technology technicians which can afford the development of new multimedia projects for the digital entertainment industry and manage content to be spread through information networks. This high-class training programme is based upon project based learning. Students learn how to build digital systems for multimedia management. Furthermore, they learn how to provide technical support to multimedia projects for cultural endeavours, telecommunications, teaching or private companies, as well as to create and provide support for all technical elements involved in sound and image creation.

IVP course is taught in third academic year, more accurate in the sixth semester. The course has a workload of 6 ECTS (European Credit Transfer and Accumulation System) distributed as 30 hours of masters classes, 30 hours of computer laboratory, and the student homework.

2.1 Teaching-Learning Methodologies

In order to perform a short review of teaching-learning methodologies used these days, there is the necessity to highlight the traditional methodology which is mainly face-to-face teaching [29] where teacher performs master classes and the students assimilate the explanations of its professor. According to this approach, one important point to be considered is the necessity to increase the motivation and focus students in order to increase their cognitive development [30, 31].

Nowadays, there are a wide range of methodologies in order to accomplish the teaching-learning paradigm in a more effective way. Let's first start with project-based learning (PBL) [3, 4, 32]. To perform this teaching methodology, teacher gives students a project to be developed along the course. It usually has milestones which can be checked in order to test students' progression in the subject. Other teaching technique is based on case-based learning [33, 34] which provides students with practical and real examples that they have to solve; usually a set of practical experiments must be resolved to pass the course. Likewise, the problem-based approach [35] has similitudes with the

previous one. Student receives problems that he has to solve; usually the problems increase their level to allow the student going deeper in the course. Otherwise, teaching-learning based on simulation and virtual environments [36, 37] allow the students to simulate a real problem or/and reduce the necessity of having multiple expensive equipment in the laboratory with the aim of teaching a course. Furthermore, the use of simulation and virtual environments is highly connected with the use social networks [38, 39] to share information and realise activities to improve knowledge of these technologies. Students can use social networks; make a wiki; and so forth. Additionally, Gamification is a teaching-learning approach that its uses is increasing nowadays [5, 40]. With this methodology students improve their skills playing video games which can be especially useful for young students.

Not exactly a methodology but as important as it is the learning platform uses in order to provide materials, documents, iteration among students. There are to consider the Learning Management System (LMS) which one of the mainly used in teaching course is Moodle. This provides a flexible environment using different types of learning methodologies [41, 42]. Furthermore, the use of 'OpenCourseWare' [43] is spreading rapidly where Universities or other companies presents a course to be followed on-line by students [44, 45]. In these courses, all the materials can be accessible thought Internet, usually utilizing some LMS.

2.2 Proposed Methodology in practical classes: Lab Sessions Setup

The course objectives for IVP are achieved by the end of each practice, students should be able to get three goals: Be familiar with the most common image processing techniques used in static image and video sequence; identify commands and functions of MATLAB to design and implement those techniques; identify and select the algorithms and methods suitable for an intended application.

The methods and techniques included in this course were selected based on the characteristics of students of a multimedia university grade. They have programming knowledge due to they studied 'Programming in C' in first course and 'Signal Processing' in second course before IVP of third course. Nevertheless, the programming knowledge is very basic by comparison with other degrees of the same university, such as computer engineer, telecommunication engineer or robotic engineer. Multimedia students usually use high-level, general-purpose and interpreted programming language to an increasing extent. More and more, the philosophy is the express programming of concepts with few lines of code.

The IVP topics studied in this course are: noise and enhancement, spatial and frequency filters, arithmetic operations with images, corner and edge detection, thresholding and segmentation, mathematical morphology, blob analysis and recognition descriptors for identification among some other.

The class are structured as four hours a week during fourteen weeks. The theoretical class are two hours of lecture in which instruments (lenses, sensors, cameras, illumination) are commented and the students learn to choose the most suitable in accordance with the target application. In addition, techniques and algorithms are presented and described, showing real applications where they can be used to solve specific problems such as [5] and [6]. The practical classes are two hours of students working in a laboratory implementing methods learned in theoretical class. Each student is provided with an iMac, MATLAB software and a short tutorial developed by teachers where the functions and commands useful for each session of practical works are briefly described and commented. This helps student to link theoretical concepts, algorithms and methods with practical commands to implement them over MATLAB. The laboratory class is designed so that the students can learn while doing. Thus, it is proposed two kinds of activities. On the one hand, four experimental blocks with several exercises each of them in which students have to process static images or very short videos applying methods commented in theoretical class. Each block has a duration of 3 or 4 weeks depending on its difficulty. In this phase, students must work individually without interaction with other students as far as possible. On the other hand, at the beginning of the course, teachers give students several video sequences with different complexity of processing. The students are organized in pairs or groups of three members. They choose two video sequences among them and they cooperate to achieve a common goal (Figure 1). Both kinds of laboratory works consisted of a progression of steps with increasing difficulty where students have to overcome the challenges proposed by teachers. Challenges are always related to real problems such as improving the quality of an image or frames of a video, detection of a patterns or objects, detection of motion and so forth. The students have to do scripts to resolve these problems using several steps of processing based on filtering, segmentation, features extraction, blob analysis, descriptors and classification by distance metrics.

The developing of the course consists of using the theoretical class so that students learn the main concepts and methods. Once the students understand those, they identify MATLAB functions and test small scripts of simple image processing algorithms as examples. Afterwards, the students have to design new scripts to resolve a similar problem but they have to design and implement their own algorithm (Figure 1). In each block, the exercises increase the difficulty in accomplishing a successful result. This is because they require that students combine more than one method or algorithm to achieve the goal.

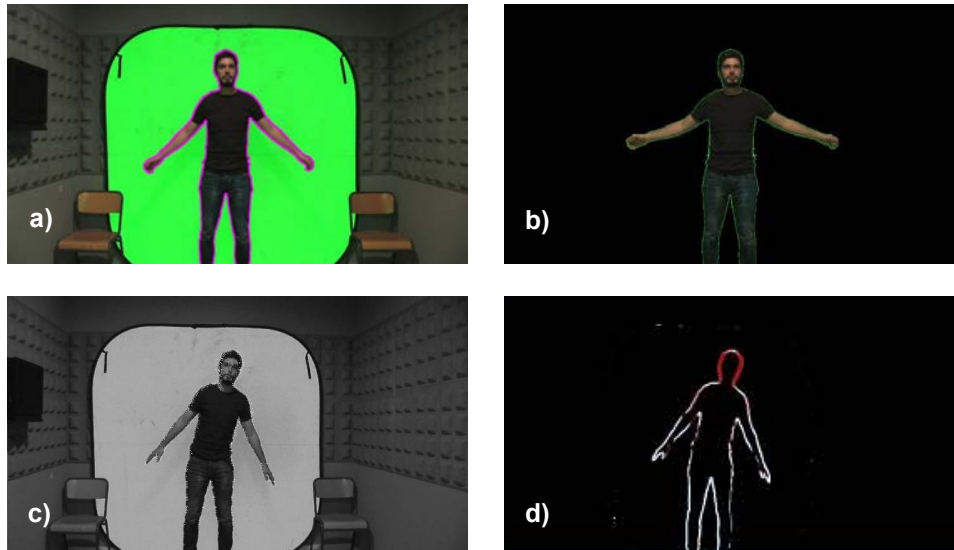


Figure 1. Solution proposed by students to detect human and its movement in a video sequence. a) Character detection within the chrome area. b) Character extraction. c) Movement Detection of the character. d) Location of the moving parts of its silhouette.

3 METHODS FOR THE ASSESSMENT OF ACQUIRED KNOWLEDGE

The ratio students/teacher is increasing on a daily basis. Thus, it is not strange to find in the literature works related with the best methodology to deal with this growing problem [46]. In [47] two alternatives were considered for the evaluation of software usage capability: group project assignment (team development of a simulation project) and an individual computer based exam. Group project mark assessment does not differentiate among any of the members of the group. As it discussed above in Section 2.1, many universities around the world are implementing project-based learning activities as an effective method to promote new learning habits [48]. Learning through this methodology cannot be assessed through traditional exams. IVP concepts require the use of a computer, and the assessment should also include this tool [47]. The European Higher Education Area (EHEA) has promoted, among others, students learning by continuous assessment, as well as the usage of virtual teaching tools like Moodle [49]. Some of these assessment methods have been developed in the IVP course at the UA.

In Section 2.2, the proposed methodology for practical sessions was described. In order to assess the practical part of the course, teachers have come up with two different assessment items: the continuous assessment through both monitoring of student progress and tests taken by the students at the end of each of the four blocks; and the report and oral exam taken at the end of the course for the students' optional project. Students have the possibility to pass the practical part of the course by passing the four tests developed by the teachers over the 14 weeks of the course. These tests are dated approximately after 3 practical sessions. The concepts asked in one test can also appear in the next one. Thus, students need to study not only the new concepts of the 3 weeks of the block, but also those taught in previous blocks of the subject.

Teachers know that learning IVP for a multimedia engineering student should be more difficult than for a computer science student because they have less programming skills. Nevertheless, they have a solid knowledge of MATLAB, and in the sessions to prepare each of the tests, they emphasize this comprehension. Teachers do not want to assess their MATLAB knowledge; this is not the aim of the IVP course. They use it as a tool to acquire all the IP and CV concepts of the subject of IVP.

Furthermore, teachers allow them to make use of MATLAB during the tests. Thus, students can use the same tools they are using in their practical sessions. Moreover, they can seek the help of program.

One of the problems generated by letting students do so is the possibility of cheating. Cheating throughout large-scale courses is one of the main problems of University courses. It becomes almost impossible for a teacher with 25 student groups to make sure that each of them is looking only at their own screen. To solve this problem, teachers have made an important effort to adapt traditional paper questions into self-corrected questions asked on a computer. The tests have been developed in a platform based on Moodle provided by the UA: Moodle UA [50]. The reason why the cheating problem is solved, or at least reduced, is that Moodle UA enables us to generate a question bank from where the test can be created. Teachers create different questions for a specific topic (called in Moodle Category). In order to build the test, the teacher can choose a number of random questions from an existing category. The idea is to randomize the questions of the students for a same group. Cheating possibilities are then reduced because students will probably not have the same test. In addition, the test can be created by shuffling randomly not only the order of the questions but also the different answers of a particular question. By entering a new question on an existing category, teachers increase the possibility of having different tests for two adjacent students.

Within the questions that a teacher can create in Moodle UA, only a few of them are valid to self-correct for IVP topics. Obviously, an essay question cannot be self-corrected. Neither can be a question where the students' answer is a drawing of an image. Teachers have developed new questions taking this into account. The concepts are always asked with the self-corrected issue in mind. In order to illustrate this idea with examples, Figure 2a shows a question prepared for the test of the third block. The answer has an error field that can be adjusted to mark as correct an answer that fits the gap between the correct area and the area considered as the right one for the image.

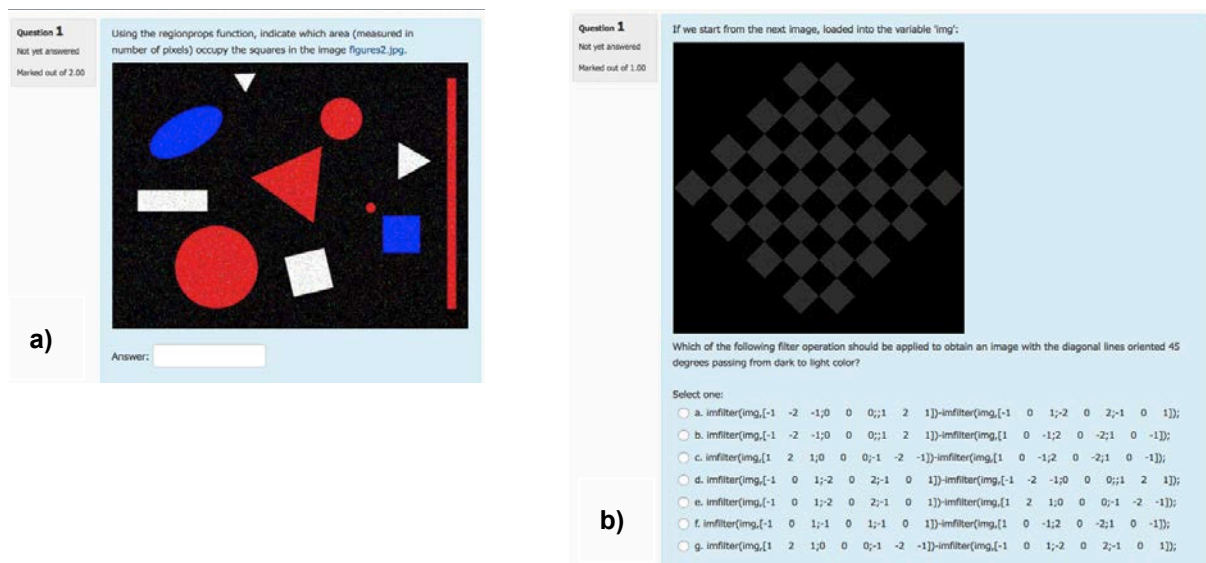


Figure 2. a) Numerical question with an error gap. b) Multiple choice question with MATLAB instructions.

Figure 2b depicts a different type of question, where the teachers offer a multiple choice display from which students must select one of them. This question can be solved by testing each of the answers in MATLAB with the image, but students who do not have a clear concept of the topic as far as the question is concerned waste a lot of time on this question and their time for the completion of the test might eventually be wasted away.

Students need to obtain a mark of 4 out of 10 in each of the four tests to pass the practical part of the course. If any of the test marks are below this threshold, the student can retake the practical fraction of the subject in a final exam. This final exam is built following the same methodology exposed previously. The bank of questions developed in Moodle UA permits to minimize the possibility of a student repeating the same questions asked during the course.

In order to assess the students' skills regarding their group working, motivation, initiative and self-learning, teachers propose an optional work at the beginning of the course. They provide the students with six videos. The students, organized in groups of 2 or 3, choose two of the videos. The idea is to indicate the movement of a person filmed in each chosen video. To achieve this, students have to do research work by themselves. The problem can be solved in different ways, but normally, they should achieve different stages (Chroma key solving, region of interest, filtering, segmentation, image analysis, etc.). The students present a report with some information about the MATLAB functions used, the stages solved, a block diagram with their proposal and the scripts generated. The teachers assess students with an online review of the videos with the student's presence. The scores depend on the computation time, robustness and accuracy in the detection of target of the proposed solution by students. Finally, in order to obtain a mark for each of the members of the group, there is an oral exam, where each of the students of the group explains any of the stages proposed to beat the challenge proposed with the videos. In addition, the students are asked by the way in which they have overcome the challenge proposed in each stage of the implemented algorithms for the videos.

4 DATA ANALYSIS AND RESULTS

In Table 1, the evaluation methodologies used for each academic year are showed. The table shows that previous academic years the evaluation was mainly done with a unique practical test of laboratory. In current year is introduced a new method of evaluation by the use of control test for each experimental block.

Table 1. Evaluation methodologies used in each academic year.

Assessment methodology / Academic years	Past course	Current course
Class Attendance	Yes	Yes
Student Monitoring in Class	Yes	No
Optional work	Yes	Yes
Final test	Yes	Yes
Partial tests as replacement of monitoring	No	Yes

From the results showed in Table 2 for the current course, still in progress, the ratio of success increases in comparison with the ratio of the past course, in which the assessment was done by a unique practical test at the end of the course and by monitoring of progression of the student. Finally, Table 3 shows two of the test results done by the students at the middle of the current course. They correspond to the first two blocks of the practical contents. It is important to remark that the final score of the practical lessons is obtained by four partial tests, i.e., one for each block, as well as the class attendance and a score of the optional work, algorithm done for human movement detection on the video sequences commented in Section 3.

Table 2. Failed and success ratio of each academic year.

Percentages Passes/Fails	Past course	Current course
> =50 %	73.6 %	86.16 %
< 50 %	26.39 %	13.85 %

Table 3. Grade acquired by students for the first and second partial test including the class attendance measured with Spanish and USA Academic Grading Systems

Percentages Hits/Faults	Spain [0-10]	USA GPA [0-4]	P1 with attendance	P2 with attendance
90-100%	Excellent (A, A+)	A (3.6-4]	13.85%	1.54%
80-89%	Very good (B+,A-)	B (3.2-3.6]	21.54 %	9.23 %
70-79%	Very good (B-,B)	C (2.8-3.2]	23.08%	4.62%
60-69%	Good (C-,C,C+)	D (2.4-2.8]	16.92%	40 %
50-59%	Sufficient (D)	E [2-2.4]	7.69%	7.69 %
< 50%	Failure (E,F)	F [0-2)	16.92%	36.92%
Student dropout rate			3%	4%
Absent			3%	3%

5 CONCLUSIONS

Students of IVP courses have difficulties in each laboratory session with the new concepts and algorithms which are presented in each laboratory practical block. The results suggested that an assessment methodology based solely on a final test which is supported in a student work monitoring in each laboratory session is not sufficient to achieve adequate learning and to measure this one. Thus, partial tests in each laboratory practical block seems more adequate and the students are more motivated for answer questions avoiding conceptual doubts and improve their level of understanding of the subject. Furthermore, the project-based learning allow student acquires new knowledge about the integration and combination of algorithms to overcome challenges as resolve real problems.

REFERENCES

- [1] Bebis, G., Egbert, D., Shah, M. (2003). Review of computer vision education. IEEE Transactions on Education 46(1), pp. 2-21.
- [2] Morison, G., Jenkins, M.D., Buggy, T., Barrie, P. (2014). An implementation focused approach to teaching image processing and machine vision- from theory to beagleboard. In Proceedings of 6th European Embedded Design in Education and Research, pp. 274-277.
- [3] Pistori, H., Pereira, M. C., Alvarez, M. A., Qi, X. (2013). Open source tools and Project-based teaching as enablers of research experience in computer vision students. In Proceedings of XLI Congresso Brasileiro de Educaao em Engenharia.
- [4] Hassner, T., Bayaz, I. (2015). Teaching computer vision: bringing research benchmarks to the classroom. ACM Transactions Computer Education, 14(4), pp. 22:1-22:17
- [5] Albarelli, A., Bergamasco, F., Torsello, A. (2011). Learning Computer Vision through the development of a Camera-trackable Game Controller. Advanced Research and Trends in New Technologies, Software, Human-Computer Interaction, and Communicability. Chapter 15, pp-154-163
- [6] Hassner, D., Serrat, J., L3pez, A. M., Baldrich, R. (2013). Traffic sign recognition for computer vision project-based learning. IEEE Transactions on Education, 56(3), pp. 364-371.
- [7] Gonzalez, Y., & Rocha, J. (2003). Some practical assignments in computer vision. In Proceedings of the Workshop on Education and Practice in Artificial Vision (pp. 101-104).
- [8] OpenCV: Open Source Computer Vision. <http://opencv.org/>, Accessed 16 May 2016.
- [9] Lux, A. (2004). The Imilab method for vision Systems. Machine Vision and Applications, 16, pp. 21-26.
- [10] RAVL: Recognition and Vision Library <http://www.ee.surrey.ac.uk/CVSSP/Ravl>, Accessed 16 May 2016.

- [11] LibCVD <http://www.edwardrosten.com/cvd/>, Accessed 16 May 2016.
- [12] Sage, D., Unser, M. (2003). Teaching image-processing programming in java. *IEEE Signal Processing Magazine* 20(6), pp. 43-52.
- [13] Sebastian, J.M, García, D., Sánchez, F.M. (2003). Remote Access Education Based on Image Acquisition and Processing through the Internet. *IEEE Transactions on Education*, 46(1), pp. 142-148
- [14] Whelan, P. F., Sadleir, R., Ghita, O. (2012). NeatVision: Development Environment for Machine Vision Engineers. In *Machine Vision Handbook*. Springer London, pp. 981-1029.
- [15] Cazorla, M., Viejo, D. (2015). JavaVis: An integrated computer vision library for teaching computer vision. *Computer Applications in Engineering Education*, 23(2), 258-267.
- [16] Gil, P., Pomares, J., Puente, S.T., Torres, F., Candelas, F., Ortiz, F.G. (2003) VISUAL: Herramienta para la enseñanza práctica de la vision artificial. In *Proceedings of the Workshop on Education and Practice in Artificial Vision*, Baleares (Spain), pp. 115-121.
- [17] Radlak, K., Frackiewicz, M., Sczepanski, M., Kawulok, M., Czardybon, M. (2015). Adaptive Vision Studio- Educational Tool for Image Processing Learning. In *Proceedings of Frontiers in Education*, El Paso (USA), pp. 1-8.
- [18] Philippsen, A., Schenk, A.D., Signorell, G.A., Mariani, V., Berneche, S., Engel, A. (2007). Collaborative EM image processing with the IPLT image processing library and toolbox. *Journal of Structural Biology* 157, pp. 28-37.
- [19] Reimer, P. Branzan, A., Tzanetakis, G. (2009). Raydiance: A tangible interface for teaching computer vision. *Lecture Notes in Computer Science*, 6939, pp. 259-269
- [20] Lomker, F., Wrede, S., Hanheide, M., Fritsch, J. (2006). Building modular vision Systems with a graphical plugin environment. In *Proceedings of the 4th IEEE International Conference on Computer Vision Systems*.
- [21] Gonzalez, R.C., Woods, R.E., Eddins, S.L. (2004). *Digital Image Processing using MATLAB*. Prentice Hall.
- [22] MATLAB: <http://es.mathworks.com/solutions/image-video-processing/>, Accessed 16 May 2016.
- [23] Kingler, T. (2003) *Image Processing with LabVIEW and IMAQ vision*. Prentice-Hall.
- [24] Jean, F., Gebali, A., Beugeling, T., Branzan-Albu, A. (2012). An educational visual prototyping environment for real-time imaging. In *Proceedings of Frontiers in Education*, Seattle (USA), pp. 1-6.
- [25] Castillo, A., Ortigón, J. (2014). Virtual laboratory for digital image processing. *IEEE Latin America Transactions*, 12(6), pp. 1176-1181.
- [26] Draper, B.A., Beveridge, J.R. (2001). Teaching image computation: from computer graphics to computer visión. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(5), pp. 823-831.
- [27] Garcia, I., Guzmán-Ramírez, E., Pacheco, C. (2015). CoLFDImaP: A web-based tool for teaching of FPGA-based digital image processing in undergraduate courses. *Computer Applications in Engineering Education*, 23(1), pp. 92-108.
- [28] Macedo, J., Colvin, K., Waldorf, D. (2005). Machine Vision Course for Manufacturing Engineering Undergraduate Students. *Journal of Manufacturing Systems* 24(3), pp. 256-265.
- [29] Iscioglu, E. (2013). Project Based Human Computer Interaction Course: An Experiment of Online and Face-to-Face Learning Environment. *International journal of engineering education*, 29(3), pp. 660-665.
- [30] Gandhi, K.I. (2016). A Teaching-Learning Model for Software Engineering Courses through Sensor-Based Cognitive Approach. *International journal of engineering education*, 32(2), pp. 915-926.
- [31] Cajiao, M.C.R., Diaz, J.A.C., Penaloza, J.T.H. ((2010). Innovation and Teamwork Training in Undergraduate Engineering Education: A Case of a Computing Engineering Course. *International journal of engineering education*, 26(6), pp. 1536-1549.

- [32] Garcia, D.F.Z., Oviedo, M.A.G., Valente, J.R.P., Barrientos, A (2016). QuadLab A Project-Based Learning Toolkit for Automation and Robotics Engineering Education. *Journal of intelligent & robotic systems*, 81(1), pp. 97-116.
- [33] Zhou, Q.G., Wu, J., Wu, T., Shen, J., Zhou, R. (2016). Learning Network Storage Curriculum With Experimental Case Based on Embedded Systems. *Computer applications in engineering education*, 24(2), pp. 186-194.
- [34] Klenk, M., Aha, D.W., Molineaux, M. (2011). The Case for Case-Based Transfer Learning. *AI Magazine*, 32(1), pp. 54-69.
- [35] Artal, J.S., Artacho, J.M., Dominguez, J.A. (2013). Integration of problem based learning with the teaching portfolio like educational tools. An experience in electrical and electronics engineering. In *Proceedings of 6th International Conference on Education, Research and Innovation (ICERI)*, Seville (Spain), pp. 1247-1256.
- [36] Scalise, K., Timms, M., Moorjani, A., Clark, L., Holtermann, K., Irvin, P.S. (2011). Student learning in science simulations: Design features that promote learning gains. *Journal of Research in Science Teaching*, 48(9), pp.1050-1078.
- [37] Smetana, L.K., Bell, R.L. (2011). Computer Simulations to Support Science Instruction and Learning: A critical review of the literature. *International Journal of Science Education*, 34(9), pp.1337-1370.
- [38] Soomro, K.A., Kale, U., Yousuf Zai, S. (2014). Pre-service teachers' and teacher-educators' experiences and attitudes toward using social networking sites for collaborative learning. *Educational Media International*, 51(4), pp. 278-294.
- [39] Minovic, M., Stavljanin, V., Milovanovic, M., Starcevic, D. (2008). Usability Issues of e-Learning Systems: Case-Study for Moodle Learning Management System. In *Proceedings of On the Move Confederated International Conference and Workshops*, Monterrey (Mexico), pp. 561-570.
- [40] Minovic, M., Garcia-Penalvo, F.J., Kearney, N. (2016). Gamification Ecosystems in Engineering Education. *International journal of engineering education*, 32(1), pp. 308-309.
- [41] Caputi, V., Garrido, A. (2015). Student-oriented planning of e-learning contents for Moodle. *Journal of network and computer applications*, 53, pp. 115-127.
- [42] Al-Balushi, S., Al-Abdali, N. (2015). Using a Moodle-Based Professional Development Program to Train Science Teachers to Teach for Creativity and its Effectiveness on their Teaching Practices. *Journal of science education and technology*, 24(4), pp. 461-475.
- [43] Abelson, H. (2008). The Creation of OpenCourseWare at MIT. *Journal of science education and technology*, 17(2), pp. 164-174.
- [44] Gil, P., Candelas, F.A., Garcia, G.J., Jara, C.A. (2012). Open Educational Resources: The Role of OCW, Blogs and Videos in Computer Network Classrooms. *International Journal of Emerging Technologies in Learning (iJET)*, 7(3), pp.4-10.
- [45] Ros, S., Hernandez, R., Read, T., Artacho, M.R., Pastor, R., Orueta, G.D. (2014). UNED OER Experience: From OCW to Open UNED. *IEEE Transactions on education*, 57(4), pp. 248-254.
- [46] Artner, N. M., Janusch, I., & Kropatsch, W. G. (2015). Evaluating and Grading Students in Large-Scale Image Processing Courses. *IEEE Computer Graphics and Applications*. 35(5), pp. 101-105.
- [47] Garcia, I., & Duran, A. (2011). Structured analysis of alternative evaluation approaches of lab sessions in engineering education. *EUROCON - International Conference on Computer as a Tool (EUROCON)*, IEEE.
- [48] Garcia, R., & Gracias, N. (2012). Project-based learning as a motivating tool to teach computer vision. *Global Engineering Education Conference (EDUCON)*, IEEE.
- [49] Chen, H. H., Lee, M. C., Wu, Y. L., Qiu, J. Y., Lin, C. H., Tang, H. Y., & Chen, C. H. (2012). An analysis of moodle in engineering education: The TAM perspective. In *Proceedings of IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, Hong Kong.

[50] Moodle UA. <http://si.ua.es/es/moodle/>, Accessed 16 May 2016.